

Ejemplo 2:

Suponga que un estudiante hace una encuesta para evaluar si los estudiantes que fuman estudian menos que los que no fuman. Los datos registrados.

REALICE UN ANÁLISIS ESTADÍSTICO DE LOS DATOS.

- 1) Activa tu directorio de trabajo.
- 2) Crea un nuevo script y llámale "Script11-DatosBivariados3"
- 3) Crea dos vectores con los datos.

```
Fuma = c("Si", "No", "No", "Si", "No", "Si", "Si", "Si", "No", "Si")
Fuma
## [1] "Si" "No" "No" "Si" "No" "Si" "Si" "Si" "No" "Si"

Cantidad = c(1, 2, 2, 3, 3, 1, 2, 1, 3, 2)
Cantidad
## [1] 1 2 2 3 3 1 2 1 3 2
```

- 4) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

```
estudia <- data.frame(Fuma=Fuma, Cantidad=Cantidad)
estudia

##      Fuma Cantidad
## 1     Si         1
## 2     No         2
## 3     No         2
## 4     Si         3
## 5     No         3
## 6     Si         1
## 7     Si         2
## 8     Si         1
## 9     No         3
## 10    Si         2

# Puedes editar los datos utilizando
fix(estudia)
```

- 5) Guarda la hoja de datos en un archivo.

```
write.table(estudia, file="Estudia.txt", append=FALSE, quote=TRUE, sep=" ",
            na="NA", col.names=TRUE)
```

6) Elimina los objetos almacenados en el área de trabajo (Workspace).

```
ls()

## [1] "Cantidad" "estudia" "Fuma"

rm(list=ls(all=TRUE))
ls()

## character(0)
```

7) Recupera desde el archivo la hoja de datos.

```
Estudia <- read.table("Estudia.txt", header=TRUE)
Estudia
```

	Fuma	Cantidad
## 1	Si	1
## 2	No	2
## 3	No	2
## 4	Si	3
## 5	No	3
## 6	Si	1
## 7	Si	2
## 8	Si	1
## 9	No	3
## 10	Si	2

8) Conecta la hoja de datos a la segunda ruta o lista de búsqueda,

```
attach(Estudia, pos=2)
search()
```

## [1]	".GlobalEnv"	"Estudia"	"package:knitr"
## [4]	"package:stats"	"package:graphics"	"package:grDevices"
## [7]	"package:utils"	"package:datasets"	"package:methods"
## [10]	"AutoLoads"	"package:base"	

9) Crea una tabla de contingencia o de doble entrada.

```
tablaCont <- table(Estudia)
tablaCont
```

```
##      Cantidad
## Fuma 1 2 3
##   No 0 2 2
##   Si 3 2 1
```

10) Calcula las tablas de proporciones o de probabilidades.

```
options(digits=3) # sólo imprime 3 lugares decimales

# Proporciones basadas en el total de la muestra, la suma de filas y columnas suman 1
propTotal <- prop.table(tablaCont)
propTotal

##      Cantidad
## Fuma   1    2    3
##   No 0.0 0.2 0.2
##   Si 0.3 0.2 0.1

# Proporciones basadas en el total por fila, cada fila suma 1
propFila <- prop.table(tablaCont, 1)
propFila

##      Cantidad
## Fuma      1      2      3
##   No 0.000 0.500 0.500
##   Si 0.500 0.333 0.167

# Proporciones basadas en el total por columna, cada columna suma 1
propCol <- prop.table(tablaCont, 2)
propCol

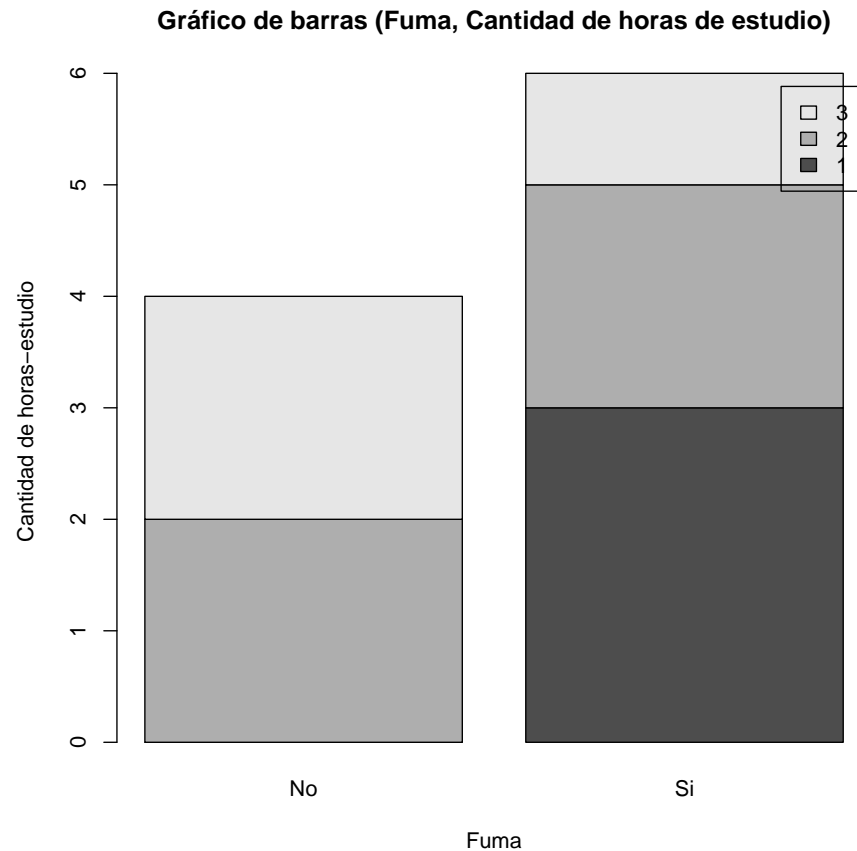
##      Cantidad
## Fuma      1      2      3
##   No 0.000 0.500 0.667
##   Si 1.000 0.500 0.333
```

11) Construya los gráficos de barras de la variable bidimensional.

```
# Gráfico de barras apiladas con la frecuencia de Cantidad como altura
barplot(table(Estudia$Cantidad, Estudia$Fuma), beside = FALSE, horizontal=FALSE,
main="Gráfico de barras (Fuma, Cantidad de horas de estudio)", legend.text =T,
xlab="Fuma", ylab="Cantidad de horas-estudio")

## Warning in plot.window(xlim, ylim, log = log, ...): "horizontal"
## is not a graphical parameter
## Warning in axis(if (horiz) 2 else 1, at = at.1, labels = names.arg,
## lty = axis.lty, : "horizontal" is not a graphical parameter
```

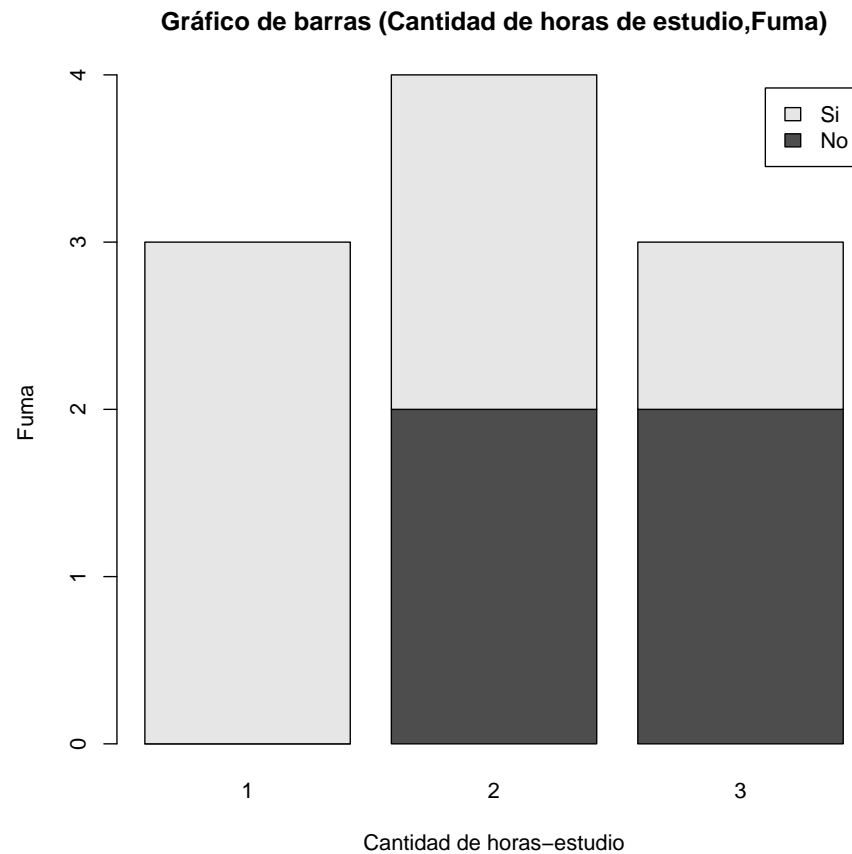
```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab,
...): "horizontal" is not a graphical parameter
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...):
"horizontal" is not a graphical parameter
```



```
# Gráfico de barras apiladas con la frecuencia de Fuma como altura
barplot(table(Estudia$Fuma, Estudia$Cantidad), beside = FALSE, horizontal=FALSE,
main="Gráfico de barras (Cantidad de horas de estudio,Fuma)", legend.text =T,
xlab="Cantidad de horas-estudio", ylab="Fuma")

## Warning in plot.window(xlim, ylim, log = log, ...): "horizontal"
is not a graphical parameter
## Warning in axis(if (horiz) 2 else 1, at = at.1, labels = names.arg,
lty = axis.lty, : "horizontal" is not a graphical parameter
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab,
...): "horizontal" is not a graphical parameter
```

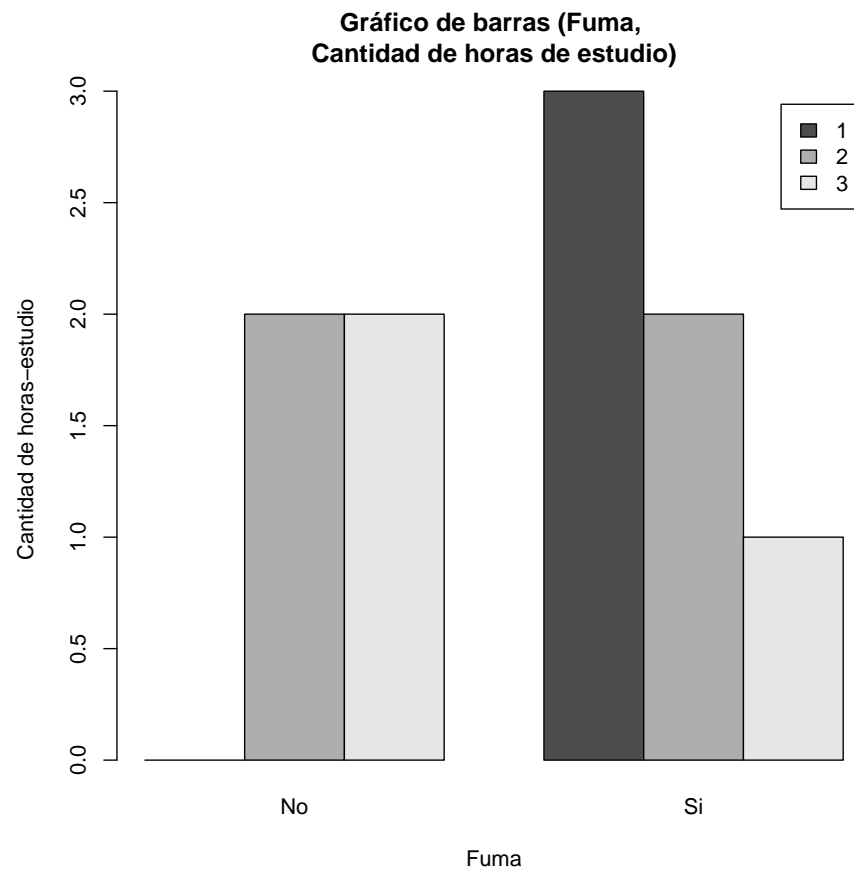
```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...):
"horizontal" is not a graphical parameter
```



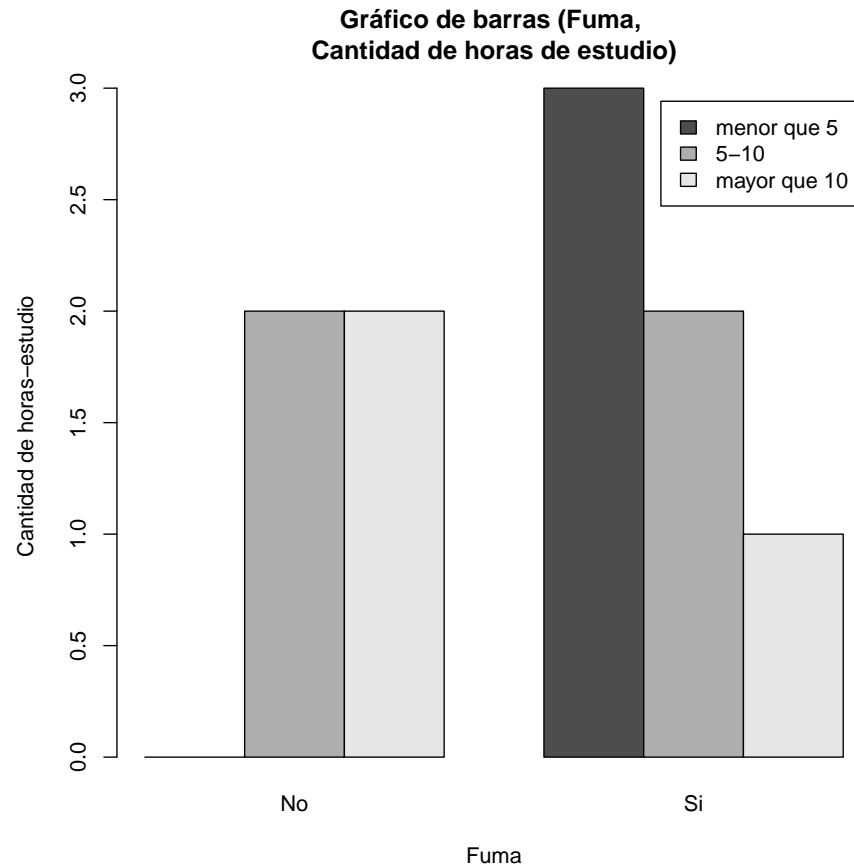
```
# Gráfico de barras no apiladas y colocación de leyenda
# Crear un factor para los nombres en la leyenda
Fuma=factor(Estudia$Fuma)
Fuma

## [1] Si No No Si No Si Si Si No Si
## Levels: No Si

barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Gráfico de barras (Fuma,
Cantidad de horas de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio",
beside=TRUE, legend.text=T)
```



```
barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Gráfico de barras (Fuma,
Cantidad de horas de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio",
beside=TRUE, legend.text=c("menor que 5", "5-10", "mayor que 10"))
```



12) Realiza la prueba o contraste Chi-cuadrado para las probabilidades dadas

```
chisq.test(tablaCont)

## Warning in chisq.test(tablaCont): Chi-squared approximation may
## be incorrect
##
## Pearson's Chi-squared test
##
## data:  tablaCont
## X-squared = 3, df = 2, p-value = 0.2

# Recuerde que las frecuencias esperadas deben ser mayores a 5 para poder utilizarlas.
# Probabilidades esperadas para la prueba Chi-cuadrada
chisq.test(tablaCont)$expected
```

```
## Warning in chisq.test(tablaCont): Chi-squared approximation may
be incorrect

##      Cantidad
## Fuma   1    2    3
##   No 1.2 1.6 1.2
##   Si 1.8 2.4 1.8
```

Sí $p - value > \alpha$ aceptar H_0 : Las variables son independientes